CLAIMS

What is claimed is:

1. A network system interconnecting a set of packet-switching network elements, the network system comprising a set of interface units, each interface unit interfacing with one of the packet-switching network elements interconnected by the network system and providing a connection of potentially variable capacity to the other interface units of the network system;

   each one of the connections capable of transporting data from its source interface unit to its destination interface unit and having an associated capacity and traffic load; the capacity of each connection controlled from its destination interface unit based at least in part on the traffic loads associated with the connections capable of transporting data to that destination interface unit.

2. The network system of Claim 1 wherein the capacity of a connection can be zero for a period of time.

3. The network system of Claim 1 wherein the traffic loads and the capacities associated with the connections between the set of interface units are dynamic variables.

4. The network system of Claim 1 where the capacities of the connections are cyclically optimized with a cycle time that is constant during regular system operation.

5. The network system of Claim 1 wherein a number, up to all, of the interface units are physically located at single physical node or platform, or are attached to the same chassis.

6. The network system of Claim 1 wherein one or more of the interface units are integrated with the packet-switching network elements they interface with.

7. The network system of Claim 1 that can be at least in part a sub-network of a multi-use or public network, with additional network elements, which do not actively participate in the operation of the thus created sub-network, potentially in pass-through mode in between of either the interface units or in between of the packet-switching network elements and the interface units of the sub-network.

8. The network of Claim 1 wherein one or more of the packet-switching network elements is another network system accordant to the definition of Claim 1, and wherein these Claim 1 networks interface with each other through regular interface units, thus allowing to cluster a number of Claim 1 networks together, potentially with a hierarchical architecture where one Claim 1 network serves as an interconnect network among a number of Claim 1 networks, thereby contributing to network scalability.

9. A network system comprising a set of logical data transport buses and a set of nodes, wherein:

each one of the logical buses has a destination node, a set of source nodes, and a capability to transport data from the source nodes to the destination node of the bus;

at each one of the buses some or all of the source nodes of the bus are on one side and the rest of the source nodes on the other side of the destination node along the logical bus, so that the destination node may divide its bus into two branches, with one bus branch on each side of the destination node for transport of data from the source nodes on that side, and wherein the furthermost source node on a bus branch from the destination node of the bus is herein called as end-of-line node of the bus branch;

each one of the nodes in the network is a destination node to one of the buses and a source node the other buses;

the individual source nodes of a bus present each a demand for a potentially variable amount of bus transport capacity;

each individual logical bus has a total data transport capacity that is divisible into a set of logical data transfer slots, a capability of being space- or timeshared among its source nodes for transport of data to its destination node via allocation of logical of the

data transfer slots on the bus among the source nodes, and a bus process cycle that is a period of time over which the total data transport capacity of the bus equals a certain number of bus data transfer slots called herein as cycle-worth of bus transfer slots; and

each individual logical bus has an at least once every bus process cycle executed control process that is capable of allocating the cycle-worth of bus transfer slots among the source nodes of the bus based at least in part on the demand for bus transport capacity presented by the individual source nodes.

10. The network system of Claim 9 that is based on the SDH/SONET standards and wherein the logical buses are formed of a group of SDH/SONET paths, wherein an SDH/SONET path can be a S11, S12, S2, S3, S4, VT-1.5, VT-2, VT-6, STS-1 or any concatenation of these, these concatenated paths including those of the form of VC-3-Nc or STS-1-Nc, wherein the suffix -Nc refers to N, an integer, number of concatenated VC-3s or STS-1s; an example of such concatenated paths of which the logical bus can be formed is STS-1-48c, a SONET path that is also often called as STS-48c or OC-48c.

11. The network system of Claim 9 wherein a set of transfer slots, even if not a continuous set, allocated to one of the source nodes of the bus form a single logical connection from the source node to the destination node of the bus, and wherein further, in case that the network system is based on SDH/SONET according to Claim 9 and thus the logical bus transfer slots are SDH/SONET paths, the single logical connection is formed via virtual concatenation of some of the SDH/SONET paths on the logical bus, and thus, in case of e.g. a bus that is made of VC-3 paths, the single logical connection would be, using the SDH standard notation, a VC-3-Xv path, wherein X, an integer, is the number of VC-3 paths concatenated to form the single logical connection.

12. The network system of Claim 9 wherein the bus control process uses a specialized control information signaling scheme to allow the logical bus to extend over multiple non-mutually-synchronized nodes and sub-networks, and to allow the nodes that

access the bus to be in any distance from each other; the signaling scheme comprising a set of sub-processes that include:

maintaining a bus process cycle timer at each one of the nodes, the timer of the end-of-line node of a bus branch called herein as master timer for the bus branch;

transmitting the bus transport capacity demand information by each source node to the destination node of the bus at least once every process cycle;

at the destination node of the bus, receiving the capacity demand information from each of the source nodes of the bus, and, before the end of its process cycle, using the newest set of source node capacity demand information as input, computing the bus transport capacity allocation information that is used to specify for each one of the source nodes on which ones of the set of bus transfer slots on the process cycle for which the table is computed the particular source node is allowed to transmit data on the bus, and that is herein considered to be presented as a set of source node -specific bus transfer slot allocation tables;

by the destination node of the bus, before the end of its process cycle, distributing the newest bus transfer slot allocation tables for the next process cycle to the source nodes of the bus, with an identifier, such as e.g. an incrementing sequence number, of the process cycle to which each table applies attached to each table;

at the source nodes of the bus, receiving the bus transfer slot allocation tables destined for it and storing the tables in a memory addressed based on the process cycle identifier of each stored table;

distributing by the end-of-line node of a bus branch the bus process cycle synchronization periodically to the downstream nodes along the bus, at a specific constant phase within every process cycle, in the form of a specially marked data structure recognized by each node accessing the bus as bus cycle synchronization pattern, which also identifies the multi-cycle process phase at the end-of-line node;

receiving and maintaining, by each of the nodes accessing the bus the cycle- and multi-cycle process synchronization of the bus as distributed by the end-of-line node of the bus branch, and thereafter applying the locally stored bus transfer slot allocation tables, each of which was computed for a particular bus process cycle, in cycle-synchronized manner by all the nodes accessing the bus.

13. The control information signaling scheme of Claim 12 wherein each one of the nodes of the network system transmits:

the transport capacity allocation information concerning the logical bus for which the particular node is the destination node;

the transport capacity demand information towards the other nodes in the network system; and

the cycle- and multi-cycle process synchronization information for the bus branches to which the node is an end-of-line node;

all within the same logical data structure, called herein as control frame, occupying a specified set of data transfer slots within each bus process cycle as defined by the node-local cycle timer, for the downstream nodes on the bus branches for which the node is an end-of-line node.

14. The control information signaling scheme as specified in Claim 13, wherein further, in case that the network system is be based on SDH/SONET according to Claim 10 and where the data transfer slots thus are time segments of SDH/SONET paths:

the bus process cycle is an integer number of SDH/SONET frame periods, the specific number of SDH/SONET frame periods in each bus process cycle known by all the nodes accessing the bus;

the cycle timer at each node counts SDH/SONET frame periods and thus:

each node transmits the control frame on a specified set of SDH/SONET frame slots within each bus process cycle on the buses for which the node is an end-of-line node, thereby enabling to maintain a synchronized copy of the bus branch master timer at each downstream node accessing the bus, so that the phase of each copy timer is kept within SDH/SONET frame period maximum phase difference from the phase of the bus master timer, thus enabling synchronized switching of SDH/SONET paths, i.e. synchronized accessing of time slots within SDH/SONET hierarchical frames, across multiple nodes, which may have independent SDH/SONET node frame phases and frequency references, along the logical data transport bus.

15. The network system of Claim 9 wherein:

each node has means to utilize during a process cycle more bus data transport capacity in total than what is the total transport capacity of any single bus or the egress capacity of any single node during a process cycle, in the network system, up to a case where a single node is able to simultaneously utilize the total data transport capacity of all the buses in the network system,

thereby enabling non-blocking line-rate multi- and broadcasting of data from a single node simultaneously to a plurality of nodes in the network system, as well as enabling to fully utilize all the transport capacity by any given source node whenever the capacity is allocated to that particular node, e.g. to unload any buffered data at the source nodes, without having to arbitrate a potentially blocking amount of total bus access capacity at the source node among the various buses on which the node may be allocated data transfer slots simultaneously.

16. A throughput maximization process for a network system interconnecting a set of packet-switching network elements, the network system comprising a set of interface units each of which interfaces with one of the packet-switching network elements and provides a connection to each other interface unit of the network, each of the connections having a source interface unit, a destination interface unit, a capability to transport a dataflow from its source interface unit to its destination interface unit, and an associated inbound dataflow volume and data transport capacity; the capacity of each connection being controlled by the network throughput maximization process associated with its destination interface unit, each such process, which thus controls the set of connections destined towards the interface unit associated with it, having a repeating process cycle comprising a set of sub-processes that include:

forwarding by the interface units sequences of packet data received from the packet-switching network elements interconnected by the network towards the destination interface units associated with each packet through a set of data buffers, wherein each one of the buffers at an interface unit is associated with one of the destination interface units reachable from that interface unit and is used for temporarily storing data of packets being forwarded towards the destination interface

unit associated with the buffer, and each buffer has a potentially variable amount of data bytes stored in it, this amount of data in the buffer called herein as the fill of the buffer;

at the interface units mapping data from each of the buffers to the connections transporting data to the destination interface unit associated with each buffer;

computing a volume of dataflow to each one of the buffers i.e. computing the inbound dataflow volumes associated with each individual connection within the network system per each process cycle;

at each of the interface units, based at least in part on the computed inbound volumes of dataflows towards it, computing an optimized set of capacities for the connections associated with these dataflows for the next process cycle;

by each interface unit, according to the computed optimal set of capacities for the connections to which it is the destination interface unit, remotely controlling these connections; and

at each interface unit, forwarding the data packets transported to it from the other interface units of the network system towards its associated packet-switching network-element over an interface that is herein called as egress interface of the interface unit and whose data transfer capacity is called as egress capacity of the interface unit,

thus the interface unit providing an optimized utilization of its egress interface capacity, and thereby the interface unit maximizing the throughput of the network system for its part, and thus the interface units of the network system together continuously with cyclic optimization processes maximizing the throughput of the entire network system interconnecting the packet-switching network-elements.

17. The network throughput optimization process of Claim 16 wherein the sub-processes of forwarding packets to the buffers associated with the destination interface units of the packets and mapping packets to the connections transporting data to the destination interface unit associated with each buffer further comprises:

a capability to classify packets based also on their priority in addition to their destination, and thereupon to store the packets of different priority classes in separate priority-class-specific segments of the buffers; and

a capability to map the packets from the priority-class-specific segments of the buffers to the connections in a prioritized order.

18. The network throughput optimization process of Claim 16 wherein the sub-process of computing the inbound volume of a dataflow further comprises the sub-steps of:

periodically sampling the fill of the buffer associated with the dataflow i.e. periodically recording a buffer fill sample;

monitoring the capacity of the connection to which the data from the buffer is mapped to, to determine data outflow volume from the buffer; and

adding the data outflow volume from the buffer, i.e. the amount of data bytes mapped from the buffer to its associated connection, between two buffer fill sampling moments to the latter of these two buffer fill samples, and subtracting from that sum the former of the two buffer fills samples,

thereby as a result of the calculation getting the volume of data inflow to the buffer during the period between the two buffer fill sampling moments, and thus, by having the two buffer fill sampling moments be one process cycle apart from each other in time, getting computed the volume of the dataflow to the buffer, i.e. the inbound dataflow volume of the connection associated with the buffer, for every process cycle.

19. The network throughput optimization process of Claim 16 wherein the sub-process of computing an optimized set of capacities for the connections that are able to transport data towards the destination interface unit associated with the process is done according to an at least once per process cycle executed computation algorithm that produces such a transport capacity allocation pattern for the next process cycle that optimizes the utilization of the egress capacity of the interface unit, the algorithm defined by a rule set as follows:

as long as there is further capacity to allocate on the next process cycle, allocate the egress capacity of the interface unit among the source interface units of the connections with that destination interface unit, until the capacity demands of the source interface units, up to even-share of process cycle worth of egress capacity per a source interface unit, are met, and after that, any unallocated capacity evenly among the source interface units whose capacity demand had not been met yet, and after that, any remaining capacity evenly among all the source interface units,

thereby maintaining optimized utilization of the egress capacity of the interface unit and enabling to route all the individual connections to the destination interface unit from each of the source interface units of the network system using transport capacity in the network system only worth of the egress capacity of the destination interface unit while allowing any one of the connections to get transport capacity in the network system even up to the full egress capacity of the destination interface unit, when the capacity demands by the individual source interface units towards the particular destination interface unit call for such capacity allocation.

20. The connection capacity optimization algorithm of Claim 19 wherein the demand for the egress capacity of the destination interface unit by a source interface unit, i.e. the demand for the capacity of the connection from that source interface unit to the destination interface unit, is computed according to a sub-algorithm that comprises a set of steps including:

comparing the fill of the buffer associated with the connection to a set of buffer fill threshold values, each of which has an associated minimum capacity demand figure, and thereby determining the minimum capacity demand for the connection, which is the minimum capacity demand figure associated with the highest of the threshold values which the buffer fill exceeded, and is herein called as buffer fill based demand figure;

determining smallest such capacity demand figure for the connection that equals or exceeds the inbound volume of the associated dataflow, this capacity demand figure called herein as dataflow volume based demand figure; and

comparing the buffer fill based demand figure to the dataflow volume based demand figure, and declaring the greater of these two as the demand for egress capacity at the destination interface unit by the source interface unit.